DDoS Attack Detection Based on RBFNN in SDN

Jingmei Li, Mengqi Zhang(✉), and Jiaxiang Wang

Harbin Engineering University, 145, Nangtong, NJ, China
happy_zmq@163.com

Abstract. SDN is a new network architecture with centralized control. By analyzing the traffic characteristics of DDoS attack, and using the SDN controller to collect the traffic in the network, the important characteristics such as the IP address entropy ratio and the port entropy ratio related to the attack are extracted. According to the analysis of relevant eigenvalues, the RBFNN algorithm is used to classify the training samples to detect DDoS attacks. Finally, the SDN environment and DDoS attacks are simulated under Ubuntu, and the RBFNN algorithm detection model is deployed in the SDN controller. Compared with BPNN algorithm and Naive Bayes algorithm, it is proved that the algorithm performs DDoS attack detection with high recognition rate in a short time.

Keywords: DDoS · SDN · RBFNN

1 Introduction

The strategy of the Distributed Denial of Service (DDoS) attacks [1] is to send a large number of seemingly legitimate network packets to the target host through a number of "zombie hosts" (hosts that have been intruded or indirectly exploited by the attacker). Finally, the target host refuses service due to network congestion or server resource exhaustion. Therefore, detecting DDoS attacks quickly and accurately has become research hotspots in the field of Internet security. Software Defined Network (SDN) [2] is a new network architecture with centralized control, programmability and hardware versatility. Network administrators can monitor network traffic in real time [3]. Comparing with traditional networks using SDN to detect DDoS attacks has the advantage of real time the global network monitoring. At the same time, the centralized control characteristics of SDN also have network security problems. When the controller is attacked by DDoS, the control plane is decoupled from the data plane, causing the entire network to collapse. Therefore, detecting DDoS attacks in SDN has important research significance and use value.

At present, researchers have proposed some detection methods for DDoS attacks under the SDN network architecture. Wang et al. proposed a DDoS attack detection method based on BPNN in software defined network [4]. This method combines OpenFlow technology to analyze the eigenvalues in the switch flow table and uses the BPNN classification algorithm to detect DDoS attacks. Fu et al. proposed a DDoS attack detection method based on KNN classification algorithm [5]. This method

S. Liu and G. Yang (Eds.): ADHIP 2018, LNICST 279, pp. 204–213, 2019.
https://doi.org/10.1007/978-3-030-19086-6_23

extracts the key features of the flow table and uses the KNN algorithm to detect DDoS attacks. Shu et al. proposed a DDoS attack detection method based on conditional entropy [6]. The method extracts the TTL and the source IP address in the switch flow table, obtains the conditional entropy of the source IP address under the same TTL value, and further analyzes the entropy change by using the sliding window non-parametric CUSUM algorithm to detect the DDoS attack. Han et al. proposed a method for detecting DDoS attacks based on entropy values [7]. The method utilizes the characteristics of the centralized control of the controller to efficiently process the information of the data packet. The DDoS attack is detected by calculating the entropy value. Based on the research and analysis of the above methods, this paper proposes a DDoS attack detection method based on Radical Basis Function Neural Network in SDN.

2 Related Technologies

2.1 Software Defined Network

The SDN architecture is a new type of network architecture [8]. The SDN architecture is shown in Fig. 1, it is to separate the control plane of the network from the data forwarding plane. The control plane calculates the forwarding rules of the network packets by the controller. The main work of the data plane is that the network equipment (such as OpenFlow switch) processes networks packets according to the forwarding rules calculated by the controller [9]. The core technology of the SDN architecture is OpenFlow technology. The OpenFlow protocol implements the flow table query, add, delete and other operations between controller and switch [10].

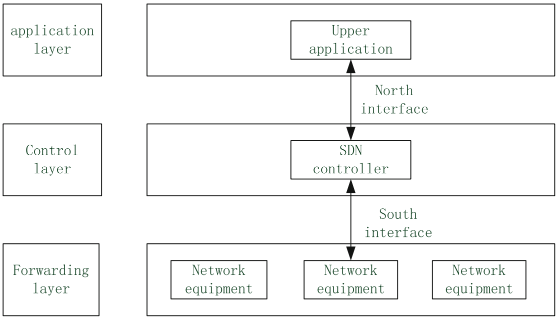

Fig. 1. SDN architecture diagram

2.2 DDoS Attacks in SDN

DDoS attacks take distributed attack, which occupies network resources through a large number of requests, causing the network paralysis and unable to provide services for legitimate users. There are controllers, switches and hosts in the SDN architecture.

According to the damage caused by DDoS attacks, DDoS attacks in SDN can be classified into three categories:

Switch denial of service. The information about the packets and the forwarding rules are stored in the flow table of the switch. When a DDoS attack occurs, a large number of false request messages are generated to occupy the flow table space of the switch. As a result, the switch can no longer allocate resources for legitimate requests.

Controller denial of service. When the information of the data packets do not match the forwarding information saved in the flow table, the controller is requested to calculate the forwarding rule of the data packets. When an attacker sends a large number of forged IP address packets, the switch will send a large number of requests to the controller, which occupies the computing and storage resources of the controller, and the controller cannot serve legitimate requests.

Host denial of service. The DDoS attacks use the defect in the host to send a large number of forged IP address requests to the target host. The computing resources, storage resources, and some other resources of the target host are occupied by the attack requests, so that the legitimate user request cannot be responded to.

3 RBFNN-Based DDoS Attack Detection

3.1 Attack Detection

RBFNN-based DDoS attack detection is divided into the following four steps:

Firstly analyze the characteristics of the DDoS attack in the SDN and the packet information in the switch flow table to determine the feature values. Using the eigenvalues of the DDoS attack dataset to train the RBFNN model to optimize the RBFNN model;

OpenFlow matches and processes network packets through user-defined or preset rules. The flow table entry structure mainly includes three parts, a header fields for packet matching, a counter for counting the number of matched packets, and an action for saving the packet processing rule. The flow table entry structure is shown in Fig. 2.

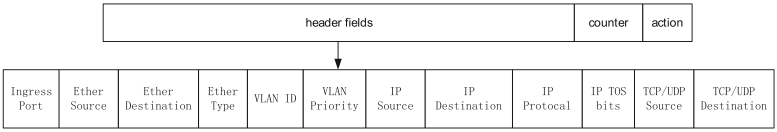

Fig. 2. Flow entry structure diagram

The DDoS attacks forge a large number of source IP addresses and sends request packets to the target host. When a DDoS attack occurs in SDN, a large number of data streams with certain regularity are generated. Because the source IP address is forged, the source IP address and source port in the data stream are scattered, and the destination IP address and destination port are concentrated. Therefore, applying IP address entropy ratio, port entropy ratio, match lookup ratio, average number of packets in per flow, and percentage of pair-flow are used as input parameters of the RBFNN.

IP address entropy ratio (EIP). When DDoS attacks occur, there are a large number of packets with forged IP addresses in the SDN. Therefore, the source IP address is more dispersed, and the destination IP address is more concentrated. The characteristics of the DDoS attack are described by calculating the entropy ratio of the source IP address to the destination IP address.

$$\mathrm{EIP} = \frac{-\sum_{i=1}^{n} p\left(\mathrm{src}_{p_i}\right) \log p\left(src_{p_i}\right)}{-\sum_{j=1}^{n} p\left(dest_{p_j}\right) \log p\left(dest_{p_j}\right)} \tag{1}$$

In the formula, src_{p_i} indicates the probability that the source IP address is p_i, and $dest_{p_i}$ indicates the probability that the destination IP address is p_{j}.

Port entropy ratio (Eport)

$$EPort = \frac{-\sum_{i=1}^{n} p(sport_i) \log(sport_i)}{-\sum_{j=1}^{n} p(dport_j) \log(dport_j)} \tag{2}$$

In the formula, $sport_i$ indicates the probability that the source port is i, $dport_j$ indicates the probability that the source port is j.

Match lookup ratio (MLR). When the switch receives data traffic, it will match flow entry. The source IP address of DDoS attacks stream is forged. Therefore, the matching rate of the switch will be drastically reduced.

$$MLR = \frac{Match}{Total} \tag{3}$$

In the formula, Math indicates the number of successful matching packets, Total indicates total number of packets.

Average number of packets in per flow (APF)

$$\mathrm{APF} = \frac{\sum_{i=1}^{Ftotal} Pnum_i}{Ftotal} \tag{4}$$

In the formula, Pnum_i indicates the number of packets in the data stream, *Ftotal* indicates the number of all packets.

Percentage of pair-flow (PCF). When DDoS attacks occur, the data flow between the attacking host and the target host does not interact, so the interactive data flow will be drastically reduced.

$$PCF = \frac{2 * PFnum}{Flownum} \tag{5}$$

In the formula, *PFnum* indicates how many pairs of interactive streams, *Flownum* indicates total number of streams.

The controller periodically sends an instruction to the switch to collect the information in the flow table;

Set the controller period to 5 s, then controller sends the ofp_flow_stats_request packet to the switch to obtain the flow table information every 5 s.

Analyzing the data information in the collected flow table to extract feature values;

After obtaining the flow table information, the controller extracts the source IP address, the destination IP address, the port number, the pairs of the interaction stream in the flow table, and the size of the data packet. The extracted related information is calculated as the eigenvalues of the RBF neural network.

Using the RBFNN model for attack detection.

The calculated eigenvalues are used as input vectors of the RBF neural network, and the RBF neural network is used to identify normal traffic and attack traffic.

3.2 RBFNN Model Training

Through the above analysis of the characteristics of the DDoS attacks, the eigenvalues are IP address entropy ratio, port entropy ratio, match lookup ratio, average number of packets in per flow, and percentage of pair-flow. The eigenvalues is calculated by the source IP address, the destination IP address, the source port number and other informations, and is used as an input unit of the RBF neural network, and the RBF neural network trains and detects the DDoS attack through the input unit. The structural diagram of the RBF neural network is shown in Fig. 3.

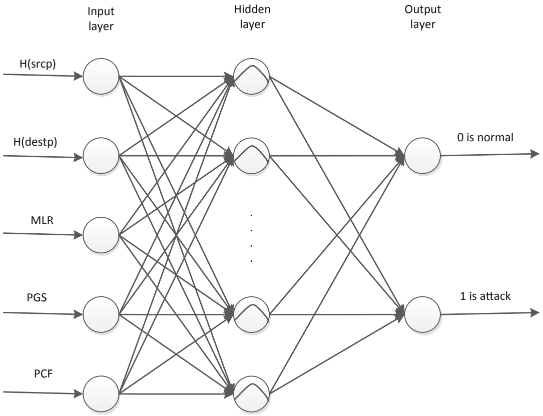

Fig. 3. RBF neural network structure diagram

RBF neural network is superior to BP neural network in terms of approximation ability, classification ability and learning speed. The training of the RBF neural network is divided into two phases:

Unsupervised learning of the hidden layer. The training of the hidden layer is to determine the parameters between the input layer and the hidden layer, and the parameters include the neuron center parameters and the corresponding width vector. The determination of the central parameters of the hidden layer neurons is a key issue in the RBF neural network. The common method for determining the central parameters is to select directly from a given training sample set according to a certain method, or to determine by clustering. In this paper, the K-means algorithm method is chosen to select the center of the hidden layer in the RBF neural network. The algorithm flow is shown in Fig. 4.

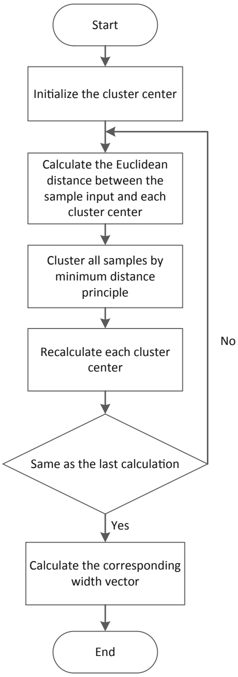
Start
Initialize the cluster center
Calculate the Euclidean distance between the sample input and each cluster center
Cluster all samples by minimum distance principle
Recalculate each cluster center
Same as the last calculation
No
Yes
Calculate the corresponding width vector
End

Fig. 4. Flow diagram of hidden layer training algorithm

The steps to calculate the center parameters using the K-means algorithm are as follows:

Set $c_{ji}(t)$ to be the central parameter of the j th hidden layer neuron for the i-th input neuron in the t-th iteration calculation, and the corresponding cluster domain is $w_{ji}(t)$.

Initialize the cluster center: Set $t = 1$, p is the total number of neurons in the hidden layer. Then the initial value of the central parameter is:

$$c_{ji}(t) = \min_i + \frac{\max_i - \min_i}{2p} + (j-1)\frac{\max_i - \min_i}{p} \tag{6}$$

In the formula, $\min_i$ is the minimum value of all input information of the i-th feature in the training set, and $\max_i$ is the maximum value of all input information of the i-th feature in the training set.

Calculate the Euclidean distance between the sample input and the cluster center $\left\|X_i - c_{ji}(t)\right\|$.

For the input samples X_i, cluster according to the principle of minimum distance: That is, if the Euclidean distance between X_i and $c_{ji}(t)$ is the smallest compared to the Euclidean distance of other cluster centers, then X_i belongs to $w_{ji}(t)$.

Recalculate the cluster centers of each type according to the classification $c_{ji}(t+1) = \frac{1}{N} \sum_{x \in w_{ji}(t)} x$.

If $c_{ji}(t+1) = c_{ji}(t)$, Return to step 2 to continue the iteration until the cluster center is unchanged.

Calculate the width vector corresponding to the central parameter of the hidden layer neuron, that is $D_{ji} = \sigma d_{ji}$, In the formula, σ is overlap coefficient, d_{ji} is the distance between the j th cluster center and other sample data centers. After determining the center parameter and the corresponding width vector, then calculating the output vector of the hidden layer based on the Gaussian function. z_j is the output value of the j-th neuron of the hidden layer, C_j is the central vector of the j th neuron in the hidden layer, D_j is the width vector of the th neuron in the hidden layer, X is input vector.

$$Z_j = \exp\left(-\left\|\frac{X - C_j}{D_j}\right\|^2\right) \tag{7}$$

Supervised learning of the output layer. The training of the output layer determines the weight between the hidden layer and the output layer. Weight training can be done by gradient descent algorithm and LMS algorithm. In this paper, the gradient descent method is used to train the weight between the hidden layer and the output layer. By adaptively adjusting the weight to the optimal value, the iterative calculation is as in Eq. 7:

$$W_{kj}(t) = w_{kj}(t-1) - \eta \frac{\partial E}{\partial w_{kj}(t-1)} + \alpha\left[w_{kj}(t-1) - w_{kj}(t-2)\right] \tag{8}$$

In the formula, $W_{kj}(t)$ represents the adjustment weight between the k th output neuron and the j th hidden layer neuron at the t-th iteration calculation. η is the learning factor and E is the evaluation function of the RBF neural network $E = \frac{1}{2}\sum_{l=1}^{N}\sum_{k=1}^{q}(Y_{lk} - O_{lk})^2$.

N is the number of hidden layer units, q is the number of output layer units, and Y_{lk} is the neural network output value of the k th output neuron at the l th input sample. O_{lk} is the expected output value of the k th output neuron at the l th input sample.

4 Experiment and Analysis

In order to verify the effectiveness of the RBFNN-based DDoS attack detection method, Under the ubuntu operating system, select Mininet to simulate the network environment, deploy the SDN with floodlight as the controller, and OpenVswitch as the switch. The network topology is shown in Fig. 5. The number 1 host is the target host, and the number 2 to number 6 host connected to the number 1 switch are normal traffic, mainly including TCP traffic, UDP traffic, and ICMP traffic. The number 6 to number 7 host connected to the number 3 switch are attacking hosts, and the TFN2K attack tool is used to generate DDoS attack traffic. The algorithm of the RBF neural network in the experiment is implemented with Python.

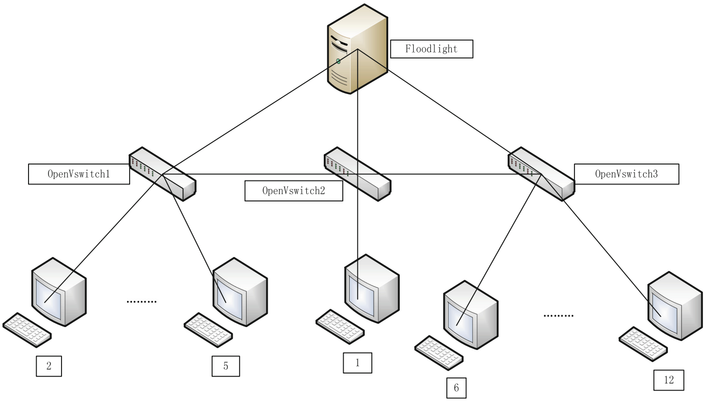

Fig. 5. Experimental network topology diagram

The controller collects the flow table information in the switch every other period, extracts the source IP address, the destination IP address and other information in the flow entry, and calculates the iput eigenvalue of the RBFNN according to the extracted information. The RBFNN algorithm classifies the traffic according to the calculated iput eigenvalue. If there is DDoS attack traffic, there is a DDoS attack in the SDN network. The results of this experiment were evaluated using two indicators: detection rate (DR) and false positive rate (FR). In the formula, TN represents the number of attack test samples that are correctly marked; FN represents the number of attack test samples that are incorrectly marked; TP represents the number of normal test samples that are correctly marked; and FP represents the number of normal test samples that are incorrectly marked.

$$\mathrm{DR} = \frac{TN}{TN + FN} \tag{9}$$

$$FR = \frac{FP}{TP + FP} \tag{10}$$

By recording and analyzing the experimental results, the detection rate and false positive rate of the DDoS attack in this experiment are shown in Table 1, and BPNN and Naive Bayes method are used as the contrast:

Table 1. Test evaluation

Training number	Test number	Naive Bayes		BPNN		RBFNN	
Nomal/DDoS	Nomal/DDoS	DR	FR	DR	FR	DR	FR
2000/2000	1000/1000	91.44%	2.11%	92.13%	1.89%	97.56%	1.97%
3000/3000	1000/1000	92.58%	2.94%	93.4%	2.71%	98.20%	2.66%
4000/4000	1000/1000	93.47%	3.52%	94.5%	3.28%	99.73%	3.24%

Compared with Naïve Bayes and BPNN algorithm, the proposed algorithm based on RBF neural network under SDN has higher detection rate and relatively lower false positive rate.

When DDoS attacks occur, the SDN controller needs to quickly detect the DDoS attacks and implement corresponding measures to prevent the DDoS attacks from causing damage to SDN. Therefore, the detection time is also an important measurement parameter of the DDoS attacks detection algorithm in the SDN. The time line diagram of each algorithm is shown in Fig. 6. It can be observed from the line diagram that as the number of samples increases, the time of RBFNN detection is stable and the time is lower. Therefore, by analyzing the detection rate, false positive rate and detection time of the experimental record results, RBFNN has a higher detection rate and detection efficiency.

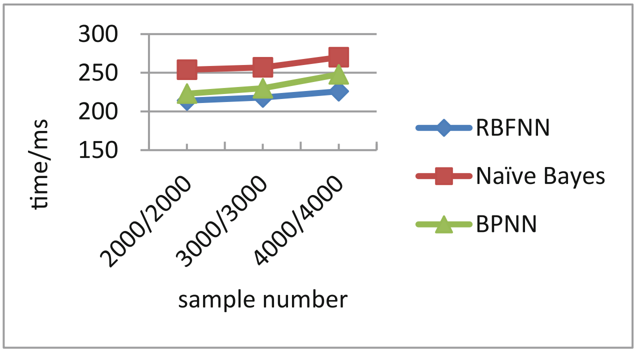

Fig. 6. The time line graph of each algorithm diagram

5 Conclusion

Through the traffic characteristics generated by the DDoS attacks and the flow table information in the switch, the five input key eigenvalues of the RBFNN are analyzed and calculated, and the RBFNN is optimally trained. The controller periodically collects flow table information, and analyzes the calculated eigenvalue IP address entropy ratio, port entropy ratio, match lookup ratio, average number of packets in per flow, and percentage of pair-flow as RBFNN inputs to detect DDoS attacks. Using the naive Bayes algorithm and BPNN algorithm as a comparison, it is proved that the RBFNN algorithm has higher detection rate and detection efficiency.

Acknowledgement. This work was supported by National Key Research and Development Plan of China (No 2016YFB0801004).

References

1. Santanna, J.J., van Rijswijk-Deij, R., Hofstede, R., et al.: Booters—an analysis of DDoS-as-a-service attacks. In: IFIP/IEEE International Symposium on Integrated Network Management, pp. 243–251. IEEE (2017)
2. Dixit, A., Hao, F., Mukherjee, S., et al.: ElastiCon; an elastic distributed SDN controller. Comput. Commun. Rev. **43**(4), 7–12 (2017)
3. Cohen, R., Lewin-Eytan, L., Naor, J.S., Raz, D.: On the effect of forwarding table size on SDN network utilization. In: Proceedings of the 33rd IEEE International Conference on Computer Communications, pp.1734–1742 (2014)
4. Wang, X., Zhuang, L., Hu, Y., et al.: DDoS attack detection based on BPNN in software defined networks. J. Comput. Appl. (2018)
5. Fu, X., Junqing, M., Xunsong, H., et al.: DDoS attack detection based on KNN in software defined networks. J. Nanjing Univ. Posts Telecommun. (Nat. Sci. Ed.) **35**(1), 84–88 (2015)
6. Shu, Y., Mei, M., Huang, W., et al.: Study on DDoS attack detection based on conditional entropy in SDN environment. Wirel. Internet Technol. **5**, 75–76 (2016)
7. Han, Z.: An entropy-based detection of DDoS attacks in SDN. Inf. Technol. **1**, 63–66 (2017)
8. Jia, W., Zhao, D., Ding, L.: An optimized RBF neural network algorithm based on partial least squares and genetic algorithm for classification of small sample. Appl. Soft Comput. **48**, 373–384 (2016)
9. Yan, Q., Yu, F.R., Gong, Q., et al.: Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: a survey, some research issues, and challenges. IEEE Commun. Surv. Tutor. **18**(1), 602–622 (2016)
10. Sahi, A., Lai, D., Li, Y., et al.: An efficient DDoS TCP flood attack detection and prevention system in a cloud environment. IEEE Access **PP**(99), 1 (2017)